

AD-A042 405

NAVAL SURFACE WEAPONS CENTER DAHLGREN LAB VA
COMPUTER MODEL FOR LIFE CYCLE COSTING. USER'S GUIDE.(U)
MAY 77 W MC LAUGHLIN
NSWC/DL-TR-3645

F/G 5/1

UNCLASSIFIED

| OF |
AD/
A042 405

NL



END

DATE

FILMED

8-77

DDC

AD A042405

NSWC/DL TR-3645

COMPUTER MODEL FOR
LIFE CYCLE COSTING
—User's Guide—

by

WAYNE McLAUGHLIN
Warfare Analysis Department

MAY 1977



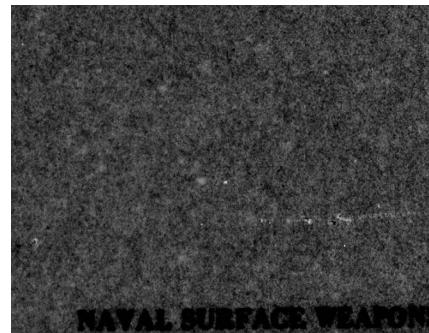
Approved for public release; distribution unlimited.

10 No.
NSWC FILE COPY



NAVAL SURFACE WEAPONS CENTER
DAEGEN LABORATORY
Dahlgren, Virginia 22448

WHITEHORN LABORATORY
Annapolis, Maryland 20740



14
NSWC/DL
-TR-
3645

92
TECHNICAL REPORT, R-3645
May 1977

9 COMPUTER MODEL FOR
LIFE CYCLE COSTING

11 May 77
12 52p.
USER'S GUIDE

by

10 Wayne McLaughlin

Warfare Analysis Department



1473
Approved for public release; distribution unlimited.

391598

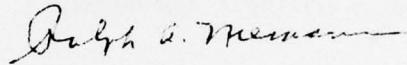
AB

FOREWORD

A computer program to model life cycle costing of equipment acquisitions has been developed and implemented at the Naval Surface Weapons Center, Dahlgren Laboratory in response to the need for an interactive, user-oriented routine which could be easily adapted to other areas of cost analysis. The requirements for the model were conceptualized at Marine Corps Development and Education Command (MCDEC), Quantico, Virginia and were communicated to NSWC by Mr. R. Sparbel of the Electronics Systems Department (DF). Mr. Sparbel was on TDY at MCDEC during model development.

This work was accomplished under project order no. T-0036 dated July 19, 1976. LCDR R. F. Rowland, Technical and Evaluation Department (DT) was the project leader. Mr. W. McLaughlin of the Computer Programming Division, Warfare Analysis Department (DK-70) was the analyst/programmer. Mr. J. C. Grey (DT), a cost analyst, contributed to the effort in the area of technical assistance.

Released by:



RALPH A. NIEMANN, Head
Warfare Analysis Department

ABSTRACT

This report provides instructions for operating the computer program which models life cycle costs for equipment acquisitions. The program, written in the Fortran IV Extended programming language, was implemented on the CDC 6700 computer under the SCOPE 3.4 operating system at the Naval Surface Weapons Center, Dahlgren, Virginia, in January, 1977.

Designed for use from an intercom terminal in an interactive manner, the model has a wide variety of built in capabilities which can greatly facilitate the task of performing life cycle cost analysis. A detailed discussion concerning the use of each feature is presented. This is then expanded into an overall description of the program's operating sequence, to aid the potential user in ascertaining if the model can be used to meet his specific needs.

The remainder of the report is devoted to the actual operation of the program. Execution involves the sequential display of requests by the terminal for input, followed by responses typed by the operator. Thorough coverage is given to the choices available to the user in each situation, including examples of input and output.

TABLE OF CONTENTS

	<u>Page</u>	
SECTION 1.	INTRODUCTION	1
1.1	Purpose	1
1.2	Model Description	1
SECTION 2.	MODEL FEATURES	5
2.1	Input Instructions	5
2.2	Families	5
2.2.1	"Systems" Costs	5
2.2.2	Multiple Families	5
2.3	Input of Cost Elements	5
2.3.1	Classes	6
2.3.2	Submitting Values	6
2.4	Overriding Cost Estimating Relationships	6
2.4.1	Transportation (R44)	6
2.4.2	Unit Production Cost (R49)	6
2.4.3	Investment Non-Recurring (R45) and Investment Recurring Costs (R46)	7
2.5	Back Up Data File	7
2.6	Data Save/Retrieval	7
SECTION 3.	SEQUENCE OF OPERATION	8
3.1	How To Access Program	8
3.2	Normal Execution	8
3.2.1	Initializations	9
3.2.2	Input Instructions	9
3.2.3	New "Class"	9
3.2.4	Data Retrieval	9
3.2.5	Run Family	11
3.2.5.1	Size	11
3.2.5.2	Multiple Families	11
3.2.5.3	Receive Input	11
3.2.5.4	"Systems" Costs	12
3.2.5.5	Compute Costs	12

TABLE OF CONTENTS

	<u>Page</u>
3.2.5.6	Output Costs 12
3.2.5.7	Reset Input Variables 12
3.2.5.8	Totals Over Family 13
3.2.5.9	Option to Save Data 13
3.2.5.10	Another Family? 13
3.2.6	Another Run? 13
3.3	Restart After Abort 14
SECTION 4.	INPUT SEQUENCE 16
4.1	Input Instructions 16
4.2	New Class 16
4.2.1	Class Name 16
4.2.2	Class Size 17
4.2.3	Members 17
4.3	Display Cost Elements? 17
4.4	Retrieve Data? 18
4.4.1	Permanent File Name 18
4.4.2	Cycle Number 18
4.5	Family Size 19
4.6	Multiple Families 19
4.7	Annotation For Output 19
4.8	Input Type 20
4.8.1	Class Identifier 20
4.8.2	Variable Name 21
4.8.3	End of Input 21
4.9	Supplying Input 22
4.9.1	Retain Value 22
4.9.2	New Value 22
4.9.3	Algorithms 22
4.9.3.1	Algorithm For No. Operators/ Equipment (R10) 22
4.9.3.1.1	Data File Used 23
4.9.3.1.2	MOS (Military Operational Specialty) . . 23

TABLE OF CONTENTS

	<u>Page</u>
4.9.3.1.3	Pay Grade 23
4.9.3.1.4	Usage 23
4.10	Save Data? 24
4.11	Another Family 25
4.12	Another Run 25
4.13	Examples of Input Sequence 26
SECTION 5.	OUTPUT SEQUENCE 32
5.1	Input Cost Element Values 32
5.2	Date & Annotation 32
5.3	Annual O & S Costs; Life Cycle Costs 32
5.4	Totals Over Family 32
5.5	Output Examples 32
SECTION 6.	SPECIAL CONDITIONS 37
6.1	Arithmetic Execution Errors 37
6.2	Phone Line Loss 37
6.2.1	Fewer Than Five Minutes 38
6.2.2	More Than Five Minutes 38
6.3	Permanent File Errors 38
6.4	Interrupting The Terminal 39
6.4.1	Deliberate Abort 39
6.4.2	Discard Current Output 39
6.5	Data Save/Retrieval Permanent File Maintenance 40
6.5.1	Audit 40
6.5.2	Purge 40
6.6	Miscellaneous Topics 41
6.6.1	Program File Already In Use 41
6.6.2	Output of Oversize Values 41
6.6.3	Error Summary 42
6.6.4	Correct Typing Errors 42

Section 1. INTRODUCTION

1.1 Purpose

Studies and analyses of life cycle costing of equipment acquisitions are being conducted at the Marine Corps Development and Education Command, Quantico, Virginia. A computer program to model life cycle costing, needed to provide assistance in this effort, was requested and has been implemented on the CDC 6700 computer at the Naval Surface Weapons Center, Dahlgren Laboratory. The program, written in the FORTRAN IV language, is executed from a remote teletype unit in an interactive manner. This report describes the computer model and supplies instructions for operating the program.

1.2 Model Description

The model uses a set of cost elements, together with associated cost estimating relationships to compute annual operations and support costs, as well as life cycle costs. The set of cost elements and estimating relationships, as set forth in reference 1, was used as a starting point for developing the model. Several new cost elements were defined and added, and some of the cost estimating relationships were modified slightly, due to specific needs of the Marine Corps and to experience and insight gained from using other computer models which already existed. However, the program user may still compute life cycle costs which conform to reference 1, by ensuring that these newly added cost elements are set to zero. The cost estimating relationships and the derivations of the cost elements may be found in reference 2. (Marine Corps TRI TAC Cost Model) The cost elements used by the present model are listed in Tables I and II.

Table I
INPUT COST ELEMENTS

<u>Cost Element</u>	<u>Definition (Units)</u>
R1	Operating hours (hrs/year)
R2	Depot Overhaul Rate (%)
R3	Transportation Cost Factor (\$/lb.)
R4	Support Equip. Maintenance Factor (%)
R5	Repair Material Cost Factor (%)
R6	Years of Operation
R7	Holding Inventory Factor (%)
R8	Power Cost (\$/kwh)
R9	Equipment Quantity (#)
R10	No. Operators/equipment (#)
R11	Annual Operator Cost (\$/yr.)
R12(1)	No. of new FSN, value greater than \$25,000 (#)
R12(2)	No. of new FSN, value - \$10,000 to \$24,999 (#)
R12(3)	No. of new FSN, value - \$2,500 to \$9,999 (#)
R12(4)	No. of new FSN, value less than \$2,500 (#)
R13	Equipment Weight (lbs.)
R14	Avg. Replacement Assembly (LRU) cost (\$)
R15	MTTR (Org. level) (hrs.)
R16	MTBF (hrs.)
R17	LRU MTTR (Int. or Deport Level) (hrs.)
R18	Unit Production Cost Est. (\$)
R19	Quantity Used for UPC Est. (#)
R20	Learning Curve Slope (%)
R21	Power Rating (kw)
R22	Preventative Maintenance (hr./yr.)
R24	Material No. 1 Consumption Rate (units/yr./equip.)
R25	Material No. 1 Cost (\$/unit)
R26	Org. Level Maintenance Personnel Cost (\$/hr.)
R27	Discard Rate (%)
R28	Int. Level Maintenance Personnel Cost (\$/hr.)
R29	Common and Peculiar Support Equip. Cost (\$)
R50	Operational Facilities (\$)
R51	Equipment Leaseholds (\$)
R52	Other Operating Costs (\$)
R53	Maintenance Facilities (\$)
R54	Contractor Services (\$)
R55	Supply Facilities (\$)
R56	Other Logistics Support Costs (\$)
R57	R & D Estimate (\$)
R58	Other Non-Recurring Investment Costs (\$)
R59	Other Recurring Investment Costs (\$)
R60	Transportation Cost Factor (%)
R61	Wt. of Avg. LRU (lbs.)
R62	Wt. of Repair Parts (lbs.)

Table I (Contd)

INPUT COST ELEMENTS

<u>Cost Element</u>	<u>Definition (Units)</u>
R63	Dist A (Org. to Int. Level) (mi.)
R64	Dist B (Int. to Depot Level) (mi.)
R65	Transportation Factor A (\$/lb./mi.)
R66	Transportation Factor B (\$/lb./mi.)
R67	Non-Recurring Investment Cost Factor (%)
R71	P2 (% of failures assigned to Int. level for repair/ discard) (%)
R72	P3 (% of failures assigned Depot level for repair/ discard) (%)
R75	Material No. 2 Consumption Rate (units/yr./equip.)
R76	Material No. 2 Cost (\$/unit)
R78	Special Materials Cost (\$)
R79	Transportation to and from Exercises (\$)
X(1)	First year cost of new FSN, value greater than \$25,000 (\$)
X(2)	First year cost of new FSN, value \$10,000 to \$24,999 (\$)
X(3)	First year cost of new FSN, value \$2,500 to \$9,999 (\$)
X(4)	First year cost of new FSN, value less than \$2,500 (\$)
Y(1)	Annual recurring cost of new FSN, value greater than \$25,000 (\$)
Y(2)	Annual recurring cost of new FSN, value \$10,000 to \$24,999 (\$)
Y(3)	Annual recurring cost of new FSN, value \$2,500 to \$9,999 (\$)
Y(4)	Annual recurring cost of new FSN, value less than \$2,500 (\$)
R80	Software Procurement Cost (\$)
R81	Software Maintenance Cost (\$/yr.)

Additionally, four pseudo-cost-elements have been added to the list of variables:

TF - Informs program how annual transportation costs (R44) are to be calculated.

UP - Informs program whether unit production cost (R49) is to be computed according to the model or overridden by the user.

INVN - Indicates whether or not the cost estimating relationship for computing investment-non-recurring costs (R45) has been overridden.

INVR - Indicates whether or not the cost estimating relationship for computing investment-recurring costs (R46) has been overridden.

Table II
COMPUTED COSTS

Annual O & S Costs

Electric Power (R33)
Special Material (R34)
Operator Personnel (R41)
Org. Maint. Personnel (R36)
Int. Maint. Personnel (R37)
Depot Maint. Personnel (R38)
Total Maint. Personnel (R35)
Support Equip. Maintenance (R39)
Software Maintenance (R81)
Supply Personnel (R42)
Spare Parts & Repair Material (R40)
Inventory Management (R23)
Inventory Holding (R68)
Inventory Administration (R43)
Transportation (R44)

Life Cycle Costs

R & D Estimate (R57)
Investment-Non-Recurring (R45)
Investment-Recurring (R46)
Total Investment (R47)
Operations (R31)
Logistics Support (R32)
Operations & Logistics Support Total (R30)
Total Life Cycle Cost (R48)

Section 2. MODEL FEATURES

2.1 Input Instructions

Instructions on how to provide input for the program are optionally available for printing out. This is the first decision that the operator must make after initiating execution of the program, and only one opportunity for obtaining these instructions is provided.

2.2 Families

The tendency to group distinct equipments together for analytical purposes results in the desire for obtaining grand totals of the life cycle costs obtained from these multiple runs. Consequently, the "family", whose members consist of the individual equipments whose cumulative total costs are to be evaluated, was chosen as the fundamental computational unit to be processed by the program. Independent runs may still be made, however, by merely viewing the family as consisting of but a single member. In this case, of course, total costs are not computed.

2.2.1 "Systems" Costs

When the size of the family being processed exceeds one, the model automatically assumes the presence of an additional "pseudo member", i.e. those costs which apply to the family as a whole but are not directly attributable to any one of its members. The user is required to submit input for these "systems" costs in addition to the other members, in the case of a multi-member family.

2.2.2 Multiple Families

If a multi-member family is being processed, it is possible to have the input cost element values saved on a temporary file, for each member of the family. This option is selected when a number of similar families, all of which have the same number of members, are to be run. Before receiving the input for the member of the current family being processed (other than the first family), the program uses the saved data to initialize the input cost elements to the corresponding member of the previous family. The life cycle costs for a slightly different family can thus be determined without undergoing the laborious process of submitting values for the entire set of cost element inputs, for every family.

2.3 Input of Cost Elements

There is no rigid sequential order in which the cost elements for a family member must be input. Depending on the situation, the user may need to supply values for all the cost elements or only a select few. This flexibility is achieved by providing the capability to submit input for any cost element individually or to submit values for particular "classes" of cost elements.

2.3.1 Classes

The input cost elements (section 1.2) are divided into six "classes" (section 4.8.1). These groupings, although somewhat arbitrary, were made so as to place those cost elements which the user might be inclined to input as a group in the same class. The classes defined are not disjoint, as some overlap does occur, but every cost input appears in at least one class. By submitting a unique identifier for the class in which he is interested, the user may input values for a group of cost elements without requesting each one individually. Additionally, a new class may be defined at execution time by the user, if he so desires.

2.3.2 Submitting Values

The program displays the current value of the next cost element to be input. The user then has the choice of keeping the present value or entering a new one. A new value may be defined by just typing the number, or in the case of some cost elements, by instructing the program that an algorithm will be used to help determine the value. The format of the response is "free-field" (in a 16-character field). If an illegal response is received, the operator is informed of the error, and given another chance to respond properly, repeatedly if necessary.

2.4 Overriding Cost Estimating Relationships

The following cost estimating relationships (reference 2) can be overridden, using the "pseudo-cost-elements" mentioned in section 1.2.

2.4.1 Transportation (R44)

The pseudo-cost-element TF, if set to a numeric value during the input, indicates that the model's cost estimating relationships for transportation (annual) are not to be used. Instead, the numeric value is taken as the analyst's estimate of transportation. TF has an additional function -- instructing the program whether to use transportation formula A or B (reference 2). If TF is set to 'A', use formula A and, if set to 'B', use formula B.

2.4.2 Unit Production Cost (R49)

The pseudo-cost-element UP may be used to override the computation of unit production cost, which is ordinarily done according to the cost estimating relationship given in reference 2. This is accomplished by defining a numeric value, which represents the analyst's estimate, for UP. Otherwise, UP is set to 'T' (tri-tac), which informs the program that the unit production cost calculation has not been overridden.

2.4.3 Investment-Non-Recurring (R45) and Investment-Recurring Costs (R46)

The model's computation of investment-non-recurring costs may be overridden by entering a numeric value for the pseudo-cost-element INVN. Similarly, the pseudo-cost-element INVR may be used to override the investment-recurring cost computation. Setting INVR or INVN equal to the character 'T' indicates that the corresponding cost is to be computed according to the cost estimating relationship given in reference 2.

2.5 Back Up Data File

In the event that execution is interrupted by loss of the phone line (for more than five minutes), or by a fatal execution error caused by data abnormality, or any of the abort conditions specified in section 3.3, the current values of the input cost elements, among other items are saved on a permanent file. This file contains all of the information necessary to resume execution (possibly after communication has been re-established) without resubmitting the input thus far provided. The exact manner in which this file is used is discussed in section 3.3.

2.6 Data Save/Retrieval

After the processing of a family is complete, the program operator may optionally save the input cost element values for the family on a permanent file for retrieval during a future run. This file will be kept in the system at NSWC/DL for a period of one year, during which the file may be accessed and the family re-run (conceivably with minor changes to the input) as many times as desired. In this manner, sensitivity analysis may be easily performed on a family over an extended period of time, because the input data for the entire family will not have to be re-entered each time.

Section 3. SEQUENCE OF OPERATION

3.1 How to Access Program

The program is resident on the permanent file disk facility of NSWC/DL's CDC 6700 computer. Communication with the computer via an intercom terminal may be achieved by dialing the telephone number obtained following proper negotiations for operating privileges with NSWC/DL. A successful connection with the computer is indicated by the display of:

NSWC/DL INTERCOM 4.4
DATE mm/dd/yy
TIME hr.mn.sec

The session with the computer must be initiated by typing:

LOGIN,XXXXXX,YYYYYY.

where XXXXXX = 7 char. user identification
YYYYYY = 6 char. computer charge code

When the terminal has been successfully logged on to the system, as indicated by the display of "COMMAND-", the following instructions (see example, section 4.13), each followed by a carriage return, will ready the program and begin execution:

1. ATTACH,MCDEC, ID=LAI.
2. ATTACH,SYSLIB.
3. LIBRARY(SYSLIB).
4. MCDEC.

The permanent file on which the program is stored is accessed by instruction #1. System - resident routines required by the program for execution are made available by commands #2 and #3. Instruction #4 tells the computer to load the program into main memory and execute it.

Under normal conditions, the message "INPUT INSTRUCTIONS? Y-YES N-NO --" will be displayed shortly after the receipt of instruction #4. The responses to this message and the others which the program subsequently displays are discussed in section 4.

3.2 Normal Execution

This section illustrates the order in which the various tasks performed by the program occur, under normal conditions. Additionally,

the various options and choices, which are made by the user, are discussed.

3.2.1 Initializations

At the outset of execution, certain program constants are assigned preset values to be retained throughout the duration of the run. These include the individual input cost element identifiers, the structure of the various classes of input cost elements, as well as which elements may be defined through the use of an algorithm, which elements may assume character values, and to what degree of precision will each element be displayed in the output section to follow, (see section 5.1). Communication to the error recovery routine is established, and the system (i.e. the computer) is notified that certain errors which may occur during the run are not to be considered fatal. Additionally, the input cost elements themselves are initialized to a set of default values, which are listed in Table III.

3.2.2 Input Instructions

If input instructions (section 2.1) are desired the (only) opportunity for obtaining them is provided next. The response concerning the instructions is the very first action taken by the operator during execution.

3.2.3 New "Class"

The capability exists whereby the user may create his own class of cost inputs at run time. This group, consisting of from 2 to 10 elements, might possibly contain, for example, the only inputs which the user wishes to change from run to run. Once established, this new class remains unchanged throughout the session since the choice of whether or not to create it can only be made once. If an abort (phone line loss, etc.) occurs after this option has been exercised, the newly defined class may be reused following a successful recovery, since all of the necessary information will have been written to the back up file.

3.2.4 Data Retrieval

A permanent file containing values for the input cost elements for a family, which has been catalogued previously by using the data saving option (section 3.2.5.9), can be retrieved for use during the current run. This enables the operator to obtain life cycle costs for a family presumably very similar (or even identical) to one which has been retained on file. If the user chooses this option, the program asks him to submit the unique permanent file name which had been submitted (section 4.10) when the family in question was saved.

After this file has been successfully identified and accessed, the number of members which the retrieved family contains is obtained by the program, followed by an initialization of the cost element inputs to

TABLE III
INITIALIZED INPUT COST ELEMENT VALUES

<u>Cost Element</u>	<u>Value</u>
R1	2920 hours
R2	20%
R3	\$.50
R4	10%
R5	5%
R6	10 years
R7	23%
R8	\$.48
R60	5%
R63	25 miles
R64	3000 miles
R65	\$.001
R66	\$.0001
R67	40%
TF	'B'
UP	'T'
INVR	'T'
INVN	'T'
all others	0

the values of the first member of the retrieved family. The number of members which the new family to be run contains must be equal to the number of members in the family on the retrieved file. Consequently, the user can not input another family size (section 3.2.5.1).

Following these initializations, the program is ready to receive input (annotation and changes) as detailed in section 3.2.5.3. The input cost elements will be initialized to those of the corresponding member of the retrieved family prior to the requests for input, for each member, until the entire family has been processed. The retrieved file is then returned to the computer system, but it may be accessed again and reused as often as desired, throughout its one year retention period.

3.2.5 Run Family

3.2.5.1 Size

At the beginning of each family, an integer representing the number of members must be submitted. The family size may vary from a single member to a maximum of ten members.

3.2.5.2 Multiple Families

If the size of the family is greater than one, the user is asked if he wishes to run multiple families. An affirmative reply causes the input cost element values to be saved on a file, over the entire family. This data is then used to initialize the cost elements for the corresponding members of the next family to be analyzed.

3.2.5.3 Receive Input

For each member of the family being processed, one line (up to seventy characters) of identification information must be supplied, for the purpose of annotating the output section.

Having submitted the annotation, the user may now enter the input values for the individual cost elements. The program initiates this by displaying a message which asks the user which type of input he wishes to make. He may respond by typing a variable name, or a class identifier.

If the response is a variable name, the program displays the current value of the variable and waits for a response. This value may be the result of the previous run, the preliminary initialization routine, or, in the case of multiple families, the value which has been reset from the corresponding member of the previous family. The user may then either enter a new value, retain the current value, or possibly invoke an algorithm to assist in defining a new value.

If the response is a class identifier (either one submitted by the user (section 3.2.3) or a program - defined class (section 4.8.1), the current value is displayed and the response is received, for each input cost element in the class, as described in the preceding paragraph.

Class identifiers and variable names may be intermixed freely during the input phase, and individual variables may be re-defined by name repeatedly if desired. However, redefining the same class of variables more than once during any input sequence is prohibited. An input type of 'END' terminates the input for the current family member being run.

3.2.5.4 "Systems" Costs

In the case of a multi-member family, the user is required to submit input for "system" costs after the final member of the family has been run, but before the grand totals are calculated. No output annotation is requested for the "systems" costs (the output is annotated, however (see section 5.2)), but the input values themselves are submitted in exactly the same way as the preceding family members (section 3.2.5.3).

3.2.5.5 Compute Costs

When the input for the family member being run is complete, the costs are computed according to the cost estimating relationships of reference 2. However, some cost computations may have been overridden, and analyst-supplied values substituted (section 2.4).

3.2.5.6 Output Costs

The output section contains a listing of the computed annual O & S costs, followed by an itemization of the life cycle costs. This is prefaced by the date of the run, the user - supplied line of annotation (section 3.2.5.3), and possibly a listing of the input cost element current values. A more detailed description of the output is presented in section 5.

3.2.5.7 Reset Input Variables

Upon completion of the running of each individual member, the input cost elements may be (1) left unchanged or (2) reset to the set of prescribed values, depending on the type of the run. If the family consists of a single member the values are left unchanged, under the anticipation that the input for the next run (presumably a one-member family, also) will be but a few modifications to the "family" which has just been completed. If a multi-member family is being run, the variables are reset to their originally initialized values at the conclusion of each individual member, under the assumption that the different members will tend to assign dissimilar values to the input cost elements. If the "multiple family" option has been exercised and the family being run is not the first such, the input cost elements are reset to the values which they had assumed

during the run of the corresponding member of the previous family.

3.2.5.8 Totals Over Family

Throughout the process of running a multi-member family, the totals for each of the output cost elements are continuously maintained. After the family run ("systems" costs included) is complete these total values over the family are printed out.

3.2.5.9 Option to Save Data

If saving the values of the input cost elements for the family just completed is desired, the option to do so may be exercised at this point. If the user elects to save the data, he is required to submit a permanent file name, which is then used by the system for identification purposes. Following a successful cataloguing operation, the file will be retained by NSWC/DL for a period of one year. The file may be used for retrieval (section 3.2.4) as many times as desired during this one-year span of retention. If it is determined, at some future time, that retrieving of this data will never take place again, the file can be permanently removed from the system (see section 6.5).

3.2.5.10 Another Family?

If the multiple family option has been selected, the program at this point asks the operator if he wishes to continue with multiple families. If the reply is "yes", the input cost element values for the first member of the family just completed are read into the appropriate locations and the request for entering output annotation is issued (section 3.2.5.3). The program then proceeds to process the family (the number of members, by necessity, must stay the same) in the manner described in sections 3.2.5.3 - 3.2.5.8. The action taken in the event of a negative reply is described in the next section.

3.2.6 Another Run?

At the conclusion of each family, after all current multiple families have been completed, the program operator is asked if he wishes to make another run. A positive response causes the program to cycle back and begin the next family by requesting the size and proceeding as in sections 3.2.5.1 - 3.2.5.9. A response of "no" causes termination of the program. The terminal will eventually display "COMMAND-". At this point, the program may be restarted from the beginning (by entering "MCDEC.") or the session may be terminated by entering "LOGOUT." The program should always be exited from in the manner just described, as opposed to just hanging up the phone, since an unwanted backup data file may be catalogued in the latter case.

3.3 Restart After Abort

The following conditions can cause the program to abort during execution:

- a. Phone line loss for more than 5 minutes. (Loss of the phone line for less than 5 minutes is discussed in section 6.2.1).
- b. Fatal arithmetic error caused by improper input data values (e.g. division by zero, taking the logarithm of a non-positive number, etc.)
- c. PP call or auto-recall error
- d. Time or storage limit exceeded
- e. Operator drop, kill, or rerun
- f. System abort
- g. CP abort

Conditions (a) and (b) will occur the most frequently, while conditions (c) - (f) will occur very seldom, if ever. When the program does abort for one of the reasons listed above, a special recovery routine is called into execution. The primary function of this routine is to write the current values of certain key variables onto a file, then catalogue this file (i.e. make it resident) in NSWC/DL's CDC 6700 permanent file system. The program employs the following strategy for using this backup file:

At the outset of any run, the program attempts to access the backup file prior to the request of any input from the operator. If this file is not found, the assumption is made that this is not an attempt to recover from one of the abort conditions (a) - (g), and execution proceeds as described in section 3.2.

However, if this file is indeed in the system, a routine which reads the file and places the data in the proper storage locations is executed. If the abort occurred during a multiple family run, another permanent file (which would have been catalogued by the recovery routine also) which contains the data from the previous family is accessed. As soon as the backup file has been read, it is purged (irreversibly removed) from the system, as a cleaning-up measure. This is indicated by a message of the form:

```
PR ID = LAI PFN=BACKUP
PR CY = 001 00000128 WORDS.:
```

Next, a display of the current values of the individual cost element inputs is provided, to enable the operator to correctly decide what his next action should be. The message requesting the next input type (section 3.2.5.3) is issued and the ensuing response, when typed by the operator, starts the program running once more. In this manner, the operator may continue inputting values at the point of interruption, or if the abort occurred after the input had been finished, simply type 'END', thereby causing the output costs to be recalculated, possibly, and the results output as usual. Execution should then proceed normally from this point on (until the next abort).

Section 4. INPUT SEQUENCE

This section discusses each of the messages which, when displayed by the program, must be responded to by the operator. The messages are issued in the order given. A carriage return is necessary to send each response to the computer.

4.1 Input Instructions

Immediately following the initiation of program execution (section 3.1, instruction #4), if an attempt to restart after an abort (section 3.3) is not being made, the message "INPUT INSTRUCTION? Y-YES N-NO --" is displayed by the terminal. Legal responses are the single characters 'Y' and 'N', which must be typed immediately after the two dashes, with no leading blanks, and followed by a carriage return. Any other response causes the message "ILLEGAL RESPONSE" to be printed. The original message is then displayed again. A response of 'Y' causes the program to list a set of instructions for supplying input before displaying the next message. No instructions are furnished if the reply is 'N', and only one opportunity to get these instructions is provided.

4.2 New Class

In addition to the "classes" of cost elements maintained by the program (sections 2.3.1, 4.8.2), a new class may be established by the user during execution. This new class, although temporary, is retained throughout the execution sequence during which it is defined. If the new class has been defined and an abort condition (section 3.3) occurs subsequently, successful execution of the recovery routine reinstates it. When the program reaches normal termination, a new class set up during the run is no longer known by the program. Only one new class may be defined and it may be done only once during each execution of the program.

The message "IS A NEW CLASS OF COST ELEMENTS TO BE CREATED? Y-YES N-NO --" asks the user if he wishes to exercise the new class option. A response of 'Y', typed after the two dashes with no leading blanks, causes the messages described in the following subsections to be displayed, assisting the user in defining the class (4.2.1 - 4.2.3). A response of 'N' refuses the option, and causes the messages described in sections 4.5 and following to be issued next. Any other response results in "ILLEGAL RESPONSE" being printed out and a subsequent re-asking of the original question.

4.2.1 Class Name

The message "INPUT CLASS NAME (1-10 CHARS, NO LEADING OR EMBEDDED BLANKS" followed by two dashes (--) at the start of the next line instructs the user to submit an identifier for the class, of his own choosing, according to the restrictions given. This will be used as a response to the message of section 4.8 whenever the operator wishes to

furnish values for the cost elements in the class. If any of the restrictions specified by the message are violated, "ILLEGAL RESPONSE" is printed and the user is given another chance to respond. The only other restriction is that the identifier must not conflict with that of one of the classes permanently contained in the program. If this is the case, "CLASS NAME CONFLICTS WITH PRESENT CLASS IDENTIFIER" is printed out, and the original message is once again displayed.

4.2.2 Class Size

The message "CLASS SIZE=" requests an integer number, at least 2 but not greater than 10, which specifies how many members the new class will contain. Leading blanks are permitted, but no decimal point is allowed. If the response is out of range, the message "SIZE MUST BE AT LEAST TWO AND NO LARGER THAN TEN" is issued, followed by another display of the original message. Any other illegal response will result in the message "X < ERROR, RETYPE RECORD AT THIS FIELD", where the character 'X' is replaced by the first character received which caused the violation. This type of error does not result in the original message being reissued, but the operator may hit the carriage return and submit another response, which will be accepted if it conforms to specifications.

4.2.3 Members

The message "INPUT COST ELEMENTS FOR CLASS, ONE PER LINE", followed by a question mark in column 1 of the next line, informs the user that the program is ready to place cost elements into the new class. The responses can be any of the input cost element identifiers (Table I, section 1.2), and the order is immaterial. No leading blanks are permitted; if the response received does not match any input cost element identifier, the message "ILLEGAL RESPONSE" is printed. If the cost element received has already been assigned to the new class, the message "COST ELEMENT MAY APPEAR IN CLASS ONLY ONCE" is displayed. In any event, question marks are displayed in column 1 of the next line and responses are received until the number of correct responses is equal to the size of the class.

4.3 Display Cost Elements?

The user must indicate whether or not he wishes to have the values of the cost element inputs displayed (section 3.2.5.6) as part of the output section, by responding to the message "IS PRINT OUT OF INPUT COST ELT. VALUES AFTER EACH FAMILY MEMBER DESIRED? Y-YES N-NO --". If the response is 'Y' the cost element values will be displayed; if 'N' this print out will be suppressed. Leading blanks are not permitted. A response other than 'Y' or 'N' produces the message "ILLEGAL RESPONSE", after which the operator is given another chance to respond properly.

4.4 Retrieve Data?

The opportunity to have the input cost elements initialized to a set of values which have been retained previously by the data saving option (section 3.2.5.9) is furnished prior to the start of the family run. The message "IS PREVIOUSLY SAVED DATA TO BE RETRIEVED? Y-YES N-NO --" may be responded to by the single character 'Y' or 'N', typed immediately after the two dashes which are printed at the end of the message. A response of 'N' indicates that no data retrieval is to take place, in which case the user is instructed to input the size of the family to be run, as detailed in section 4.5. If the response is 'Y', the user must identify the data he wishes to retrieve by specifying the permanent file on which it resides. The method of doing this is discussed in sections 4.4.1 and 4.4.2. A response different from the two given above causes the display of "ILLFGAL RESPONSE". The user is then given another opportunity to submit an acceptable response.

4.4.1 Permanent File Name

Having selected the retrieval option, the user must submit the name of the data file he wishes to use. This is accomplished by responding to the message "ENTER PERMANENT FILE NAME (1-39 CHARACTERS, NO LEADING OR EMBEDDED BLANKS) --". The response, which must be in compliance with the conditions specified by the message, is typed immediately after the two dashes (--). A further restriction is that the response must contain only letters and numbers, i.e. special characters such as '+', '-', '*', ':', etc., could not possibly have been used to name the permanent file. Additionally, the reserved permanent file names (section 4.10) must not be used. Unacceptable responses are indicated by the message "ILLEGAL RESPONSE". The user is instructed to re-enter the permanent file name when this occurs.

4.4.2 Cycle Number

When a file of data is saved permanently, it is automatically assigned a cycle number by the system. This integer number, whose range of values is from 1 to 999, enables more than one file to be saved under the same permanent file name in the following manner:

A permanent file saved under a new name (one which does not currently exist in the system) is given the cycle number of 1. Thereafter, each file which is cataloged under the same permanent file name is assigned a cycle number equal to one greater than the highest cycle which is present. Thus, the second file would be given the cycle number of 2, the third 3, and so on. A maximum of five distinct cycles may co-exist under the same permanent file name.

This capability necessitates that the user specify a cycle number when he wishes to retrieve any data file, by entering this number in response to the message "CYCLE NO. =". The response may be typed any-

where after the = sign. Leading blanks are allowed but a decimal point is not allowed. The number entered must be greater than 0 and less than 1000; if not, the message "ILLEGAL CYCLE NUMBER" is displayed. The operator will then be given another opportunity to respond. All other unacceptable responses produce the message "X < ERROR, RETYPE RECORD AT THIS FIELD". The character 'X' in the message will be replaced by the character which caused the error. When this type of error occurs, the user must resubmit the cycle number, following a carriage return.

The permanent file name and cycle number combination having been submitted properly, an attempt is made to access the file. If it is found, execution proceeds. If a file has not been saved under the identifiers submitted, the operator is so informed by a permanent file error message (section 6.3) with an error code of 12 octal. Another permanent file name and cycle number will then be requested.

4.5 Family Size

The next message, "FAMILY SIZE =", requests an integer which represents the size of the family to be run. The size must be at least 1 and at most 10. Leading blanks are permitted but non-numeric characters are illegal. The error message "FAMILY SIZE OUT OF RANGE" informs the operator that the number entered is less than 1 or greater than 10. The original message is reissued in this case. Other unacceptable responses result in the message "X < ERROR, RETYPE RECORD AT THIS FIELD", where the character 'X' is replaced by the first character which caused the error. This error does not cause the original message to be reissued, but a carriage return followed by a legal value will correct the situation.

4.6 Multiple Families

If the family contains more than one member, the message "MULTIPLE FAMILIES? Y-YES N-NO --" is displayed. If the user wishes to take advantage of the data saving feature described in section 2.2.2, he responds by typing 'Y' (after the two dashes (--)). A response of 'N' means that the option is not desired. Again, leading blanks are not acceptable, and invalid characters will produce the message "ILLEGAL RESPONSE", then a redisplay of the original message. The issuance of the multiple family message is by-passed if the family contains only one member.

4.7 Annotation for Output

The message "ENTER ANNOTATION FOR OUTPUT" is followed by a question mark in column 1 of the next line. The user may then enter one line (up to 70 characters) of identification information of his own choosing for this family member. As implied by the message, this information will be used as a heading for the print out of the results. This message is not displayed when the "systems" costs for the family (section 2.2.1) are being input. Instead, "INPUT SYSTEM COST ELEMENTS" is displayed. There is no

response required to this message, and it is followed immediately by the message described in the next section.

4.8 Input Type

The message "ENTER INPUT TYPE (CLASS ID, VAR NAME, OR 'END')" requests that the operator specify what type of input he wishes to make. The response, which directly follows the question mark printed in column 1 of the next line, must be one of the types specified in sections 4.8.1 - 4.8.3. Leading blanks are unacceptable; this response and any other which does not conform to the three types to follow causes "ILLEGAL INPUT TYPE", followed by a redisplay of the original message, to be printed.

This message is displayed one time per family member input, but may be responded to any number of times. A question mark is printed in column 1 of the next line, signaling that the program is ready for the next input type, after the input sequence generated by the last legitimate input type submitted is complete.

4.8.1 Class Identifier

An input type of a class identifier, either that of one of the classes which are permanently stored in the program or of the new class previously defined by the user, means that values are to be supplied for all of the cost elements of that particular class. The class identifiers currently in the program and the cost elements contained in each are listed in Table IV.

Table IV
CLASS IDENTIFIERS AND COST ELEMENTS

<u>OPS (Operations)</u>	<u>MNT (Maintenance)</u>	<u>SUP (Supply)</u>
R9	R9	R9
R10	R13	R12(1)
R11	R15	R12(2)
R21	R16	R12(3)
R24	R17	R12(4)
R25	R22	R14
R50	R26	R16
R51	R27	R27
R52	R28	R55
R75	R29	R56
R76	R53	R81
	R54	

Table IV (Cont.)
CLASS IDENTIFIERS AND COST ELEMENTS

<u>TRB (Transportation 'B')</u>	<u>RDI (RD & I)</u>	<u>PCN (Program constants)</u>
R9	R9	R1
R16	R18	R2
R61	R19	R3
R62	R20	R4
R63	R57	R5
R64	R58	R6
R65	R59	R7
R66	UP	R8
R71	R78	R60
R72	R80	R67
TF	INVN	X(1)
R79	INVR	X(2)
		X(3)
		X(4)
		Y(1)
		Y(2)
		Y(3)
		Y(4)

The cost elements for the desired class are input sequentially, in the order shown. If two different classes having one or more cost elements in common are requested by entering each of their class identifiers during the input phase of a single family member, the overlapping member(s) can not be input a second time. It is not possible to submit values for the cost elements of a class more than once during the input of a single family member by reentering the class identifier. An attempt to do this results in the input type response being ignored. However, a cost element may be redefined individually as many times as desired (section 4.8.2).

4.8.2 Variable Name

A (possibly new) value may be defined for any of the cost element inputs (Table I, section 1.2) individually by entering its associated identifier. The same cost element may be redefined individually as many times as desired during input by resubmitting its identifier.

4.8.3 End of Input

The word 'END', entered as an input type, terminates input for the family member being run. This causes the life cycle costs to be calculated, and the results to be printed out.

4.9 Supplying Input

Prior to accepting a new value for a cost element, whether its identifier has been entered individually or whether it is a member of a class which has been requested, the program displays the current value according to the following format: ID ... = XXXXXXXX.XX = , where ID represents the cost element identifier and the X's represent the current value, expressed to two decimal places. The response is typed anywhere in the sixteen character field immediately following the second equals sign. Leading blanks are allowed and numeric responses need not contain a decimal point, but embedded blanks are illegal. The three proper response types are discussed in section 4.9.1 - 4.9.3. If the response is not of these three types, the message "ILLEGAL RESPONSE" is displayed. This is followed by a redisplay of the current value of the variable in question, as discussed at the beginning of this paragraph.

4.9.1 Retain Value

The operator may retain the current value of the cost element as displayed by responding with the single character 'P'.

4.9.2 New Value

Responding with a number other than the one displayed causes the cost element to be set to that value. The "pseudo" cost elements, which can assume either numeric or character values, may be redefined to any "allowable" character (sections 2.4.1, ff) by responding with the character. Cost elements which are percentages must be entered as whole numbers and not as decimals (e.g. 75% would be entered as 75.0 and not .75).

4.9.3 Algorithms

Special routines, called algorithms, exist in the model for particular input cost elements. In general, these routines use operator responses to locate values in a table (contained on a file), thereby producing an estimate for the cost element in question. A response of 'A' causes the appropriate algorithm to be executed. Typing 'A' for a cost element which does not have an associated algorithm will produce the message "NO ALGORITHM EXISTS FOR xx", where xx is the cost element identifier. Sections 4.9.3.1, ff. discuss the individual algorithms in detail.

4.9.3.1 Algorithm for No. Operators/Equipment (R10)

This routine actually defines values for R10 and annual operator cost (R11) jointly. However, the algorithm can not be invoked by a response of 'A' for R11. The operator cost is determined by looking up (MOS, pay grade) combinations in a permanent file manpower cost data base and multiplying this value by an input usage (# of men and fractions thereof), for as many combinations of MOS, pay grade, and usage as de-

sired. Upon exit from the routine, the sum of all such products is divided by the sum of the usages input and this average is placed in R11. R10 is set to the sum of the usages. The responses which must be supplied by the operator of the program are discussed in the following sections.

4.9.3.1.1 Data File Used

There are two different files of manpower cost data available to this algorithm. The operator selects the one he wants by responding to "ENTER MANPOWER COST FILE DESIRED 1 -- TRI TAC 2 -- COL EVANS". The response of '1' or '2' is entered on the next line, on which two dashes (--) will have been displayed at the beginning. No leading blanks are allowed. Errant responses produce the message "ILLEGAL RESPONSE" followed by another chance. If a problem in accessing the data file occurs, a message (section 6.3) is issued and control returned to the main program.

4.9.3.1.2 MOS (Military Operational Speciality)

The message "MOS =" requests that the desired MOS be entered. Leading blanks are illegal. A response of 'P' terminates the algorithm, resulting in a display of the estimated average cost (R10) and total usage (R11) obtained from previous (MOS, pay grade, usage) responses, followed by a return to the current input sequence.

The MOS entries in the data file are searched for a match to the response received. If one is found, the algorithm continues. Otherwise, the message "DATA ON THIS MOS NOT AVAILABLE" is printed, "MOS =" is displayed again, and another response is accepted. This cycle continues until the MOS submitted coincides with one of those on the file.

4.9.3.1.3 Pay Grade

Having supplied the MOS, the operator must next input the associated pay grade. The message "PAY GRADE =" may be responded to with no leading blanks, by one of the following: 'E1', 'E2', ..., 'E9'. "ILLEGAL RESPONSE" followed by a redisplay of the original message is the result of typing any other combination of characters.

4.9.3.1.4 Usage

The number of men associated with the previously supplied (MOS, pay grade) combination is required next, as indicated by the display of "USAGE =". The response may be any positive number, and may include decimal places. The decimal point is optional if the usage is a whole number, and leading blanks are allowed. If a non-positive number is entered, the message "ILLEGAL RESPONSE" is displayed, then "USAGE =" again. Non-numeric characters will cause "X < ERROR, RETYPE RECORD AT THIS FIELD", where X represents the first non-numeric character received by the computer. In this case, a carriage return followed by a legal response corrects the error. After obtaining the usage, the program recycles back to the message "MOS =", where another (possibly different) MOS can be entered, or

the algorithm terminated (section 4.9.3.1.2).

4.10 Save data?

At the conclusion of each family run, the data (current values of input cost elements) may be saved on a file for future retrieval. The desire to save this data is indicated by a response of 'Y' to the message "SAVE DATA? Y-YES N-NO --". A response of 'N' informs the program that the data for this family is not to be saved. The response (no leading blanks) must be either 'Y' or 'N'; otherwise, the message "ILLEGAL RESPONSE" is displayed, followed by a redisplay of the original message.

If the operator chooses to have the family data saved, he must submit a name for the permanent file which will retain this data. This is done in response to the message "ENTER PERMANENT FILE NAME (1-39 CHARACTERS, NO LEADING OR EMBEDDED BLANKS) --". In addition to conforming to the restrictions indicated by the message, the response must contain only letters and numbers. Special characters, e.g. '+', '-', '*', ':', etc., will not be accepted. The following permanent file names have been reserved for use by the program, and cannot be used to name a saved file: BACKUP, MULTFAM, SAVE, MCDEC (or NAVELEX), MAN, and EVANS. A response which does not meet the specifications detailed above produces the message "ILLEGAL RESPONSE". The operator must then submit another permanent file name.

After the saved file has been successfully named, the program attempts to perform the function of entering it into the system. If the cycle number limit of 5 (section 4.4.2) has been reached, a permanent file error message (section 6.3) is issued with an error code of 4. The operator will then be instructed to enter another permanent file name.

A successful saving of the data is indicated by the display of a message of the form:

```
CT ID =      LAI PFN=xxxxx...
CT CY = nnn 00000512 WORDS.:
```

where

xxxxx.... - the name which the user has assigned to the file
nnn - the cycle number assigned to the file.

The cycle number is the integer which the user must submit when retrieving the file (section 4.4.2). The system assigns this number to the file in order to be able to differentiate among different files saved under the same permanent file name. When a name is submitted initially, the resulting file is given a cycle number of 1. Each succeeding file (with the same name) is assigned a cycle number equal to one greater than that of the highest existing cycle. It is thus possible to save up to five data sets under the same permanent file name.

4.11 Another Family

The message "ANOTHER FAMILY? Y-YES N-NO --" is issued after the completion of output of results of the family being run only if the following conditions hold:

1. The family just completed has more than one member.
2. The multiple family option (section 4.6) has been selected.

The response must be a single character, either 'Y' or 'N', typed after the two dashes (--) with no leading blanks. The program will display "ILLEGAL RESPONSE" if any other character(s) is submitted, then reissue the original message.

A response of 'Y' informs the program that another family is to be run with the multiple family option retained. This results in the input cost elements being initialized via the saved multiple family data file to the values supplied for the first member of the family just finished, and the program cycling back to display the message discussed in section 4.7. The input data for this family will be saved for use in the event that still another multiple family is to be processed.

A response of 'N' deselects the multiple family option, thereby informing the program that this series of families has been completed. If an abort has occurred and the program has been restarted, the multiple family file will have been made permanent and thus must be removed from the system. The successful removal of the file is indicated by a message of the form:

```
PR ID =      LAI  PFN=MULTFAM
PR CY = 001      00001024 WORDS.:
```

4.12 Another Run

The message "ANOTHER RUN? Y-YES N-NO --" is displayed following a response of 'N' to the message of section 4-11 or if the current family has been finished and it is inappropriate to issue the message of section 4.11. The response is typed after the two dashes (--) . The two legal responses, 'Y' and 'N', must not contain leading blanks. The 'Y' response causes a recycling back to the display of section 4.5, where the next family is begun. A response of 'N' terminates the program. Any other response causes "ILLEGAL RESPONSE" to be printed, then the original message redisplayed.

4.13 Examples of Input Sequence

The following examples demonstrate the sequence of input during the program's operation. Terminal displays and operator responses are intermixed throughout, thus illustrating the interactive nature of the program. The responses necessary to initiate execution, after the user has logged in, are shown at the beginning of the first example.

Example one makes use of the following program features and options:

a. A new class of input cost elements is created for use during the current execution cycle only.

b. The display of the input cost element values prior to the computed costs is suppressed.

c. Input cost element values are supplied by individual request as well as through the program-defined "classes".

d. The data is saved on the permanent file "EXAMPLE" for later retrieval.

e. Another run is made after a slight modification. The new class created at the beginning of the execution cycle is used to make the desired changes.

The output life cycle costs produced by this example are given in section 5.5.

Example two uses the retrieval option to access the data saved during example one. Another set of life cycle costs was computed after the following modifications were made:

a. The cost element R9 (equipment quantity) was changed to 100.

b. The pseudo-cost-element UP was used to override the model's unit production cost computation.

c. The investment-non-recurring cost computation was overridden by assigning a value of \$1000000 to the pseudo-cost-element INVN.

The option to display the input cost element values was selected for this example. The output which resulted, including the life cycle costs, is presented in section 5.5.

Example 1

COMMAND- ATTACH,MCDEC, ID=LAI.
PFN=
MCDEC
PF CY. NO.=003
COMMAND- ATTACH,SYSLIB.
PFN=
SYSLIB
PF CY. NO.=076
COMMAND- LIBRARY(SYSLIB)
COMMAND- MCDEC.

BEST AVAILABLE COPY

INPUT INSTRUCTIONS? Y-YES N-NO --N

IS A NEW CLASS OF COST INPUTS TO BE CREATED? Y - YES N-NO --Y

INPUT CLASS NAME (1 - 10 CHARS, NO LEADING OR EMBEDDED BLANKS)
--NEWCLASS

CLASS SIZE =3

INPUT COST ELEMENTS FOR CLASS, ONE PER LINE

?R9

?R20

?TF

IS PRINT OUT OF INPUT COST ELT. VALUES AFTER EACH FAM. MEMBER DESIRED?
Y - YES N - NO --N

IS PREVIOUSLY SAVED DATA TO BE RETRIEVED? Y-YES N-NO --N

FAMILY SIZE =1

ENTER ANNOTATION FOR OUTPUT

?EXAMPLE 1-A

ENTER INPUT TYPE (CLASS ID, VAR NAME, OR 'END')

?R5

R5 = 5.00=15

?R9

R9 = 0.00=234

?R18

R18 = 0.00=30000

BEST AVAILABLE COPY

?R19
R19 = 0.00=234
?R57
R57 = 0.00=1000000
?TRB
R9 = 234.00=P
R16 = 0.00=1000
R61 = 0.00=5
R62 = 0.00=30
R63 = 25.00=P
R64 = 3000.00=P
R65 = .00=0.05
R66 = .00=0.01
R71 = 0.00=P
R72 = 0.00=5
TF = B=P
R79 = 0.00=P
?SUP
R12(1) = 0.00=25
R12(2) = 0.00=25
R12(3) = 0.00=25
R12(4) = 0.00=25
R14 = 0.00=1000
R27 = 0.00=.1
R55 = 0.00=P
R56 = 0.00=P
R81 = 0.00=P
?MNT
R13 = 0.00=30
R15 = 0.00=3.123
R17 = 0.00=20
R22 = 0.00=P
R26 = 0.00=P
R28 = 0.00=5
R29 = 0.00=5000
R53 = 0.00=P
R54 = 0.00=P
?DPS
R10 = 0.00=3
R11 = 0.00=12000
R21 = 0.00=0.05
R24 = 0.00=100
R25 = 0.00=3
R50 = 0.00=P
R51 = 0.00=P
R52 = 0.00=P
R75 = 0.00=P
R76 = 0.00=P
?END

BEST AVAILABLE COPY

(results, see section 5.5)

SAVE DATA? Y-YES N-NO --Y

ENTER PERMANENT FILE NAME (1-39 CHARACTERS, NO LEADING OR EMBEDDED BLANKS) --EXAMPLE

ANOTHER RUN? Y-YES N-NO --

CT ID= LAI PFN=EXAMPLE:

CT CY= 001 00000512 WORDS.:

Y

IS PREVIOUSLY SAVED DATA TO BE RETRIEVED? Y-YES N-NO --N

FAMILY SIZE =1

ENTER ANNOTATION FOR OUTPUT

?EXAMPLE 1-B

ENTER INPUT TYPE (CLASS ID, VAR NAME, OR 'END')

?NEWCLASS

R9 = 234.00=300

R20 = 0.00=75

TF = B=A

?END

(results, see section 5.5)

BEST AVAILABLE COPY

SAVE DATA? Y-YES N-NO --N

ANOTHER RUN? Y-YES N-NO --N

END LCC

.939 CP SECONDS EXECUTION TIME

BEST AVAILABLE COPY

Example 2

COMMAND- MCDEC.

INPUT INSTRUCTIONS? Y-YES N-NO --N

IS A NEW CLASS OF COST INPUTS TO BE CREATED? Y - YES N-NO --N

IS PRINT OUT OF INPUT COST ELT. VALUES AFTER EACH FAM. MEMBER DESIRED?
Y - YES N - NO --Y

IS PREVIOUSLY SAVED DATA TO BE RETRIEVED? Y-YES N-NO --Y

ENTER PERMANENT FILE NAME (1-39 CHARACTERS, NO LEADING OR EMBEDDED
BLANKS) --EXAMPLE

CYCLE NO. =1

ENTER ANNOTATION FOR OUTPUT

?EXAMPLE 2

ENTER INPUT TYPE (CLASS ID, VAR NAME, OR 'END')

?R9

R9 = 234.00=100

?UP

UP = T=50000

?INVN

INVN = T=1000000

?END

(results, see section 5.5)

SAVE DATA? Y-YES N-NO --N

ANOTHER RUN? Y-YES N-NO --N

END LCC

.368 CP SECONDS EXECUTION TIME

COMMAND- LOGOUT.

CPA 6.419 SEC. 3.851 ADJ.

CPB .000 SEC. .000 ADJ.

UU 8.873

CONNECT TIME 0 HRS. 32 MIN.

05/02/77 LOGGED OUT AT 14.18.35.

<

Section 5. OUTPUT SEQUENCE

The results of the processing of a family member are displayed upon the response of 'END' to the message described in section 4.8. The following sections discuss the individual items which are printed.

5.1 Input Cost Element Values

The first portion of the output is a display of the values of the cost element inputs in tabular form. The values given for the cost elements are those used for computing the cost estimates for the current family member. As indicated in section 4.3, these cost elements may optionally be excluded from the output section.

5.2 Date and Annotation

The current date, in the form dd/mm/yy is printed. This is followed by the line of annotation which has been entered in response to the message of section 4.7. In the case of a multi-member family, when the "system" costs are being output, the annotation consists of the line ""SYSTEM" COSTS FOR FOLLOWING FAMILY:" followed by a line by line listing of the annotation which had been input previously for this family.

5.3 Annual O & S Costs; Life Cycle Costs

The computed costs for the current family member, as given in Table II, section 1.2, are listed next. The appropriately titled costs are given in dollars for annual and thousands of dollars for the life cycle.

5.4 Totals Over Family

Upon completion of the output for the "systems" costs of a multimember family, the program displays the grand totals over the family. This portion consists of the current date and the line "TOTAL COSTS OVER FAMILY:", which is followed by a line by line listing of the annotation which had been entered for each of the family members (excluding "systems" costs). The total annual costs and the total life cycle costs are then printed, according to the format described in section 5.3.

5.5 Output Examples

Samples of the results produced by the program are presented in this section. The input sequences responsible are shown in section 4.13.

BEST AVAILABLE COPY

ANNUAL O & S COSTS IN \$

ELECTRIC POWER	16398.72
SPECIAL MATERIAL	70200.00
OPERATOR PERSONNEL	8424000.00
ORG. MAINT. PERSONNEL	0.00
INT. MAINT. PERSONNEL	68259.67
DEPOT MAINT. PERSONNEL	334476.06
TOTAL MAINT. PERSONNEL	402735.73
SUPPORT EQUIP. MAINT.	500.00
SOFTWARE MAINTENANCE	0.00
SUPPLY PERSONNEL	2047.79
SPARE PARTS & REPAIR MATERIAL	103072.79
INVENTORY MANAGEMENT	38250.00
INVENTORY HOLDING	230336.63
INVENTORY ADMINISTRATION	268586.63
TRANSPORTATION	112100.63

LIFE CYCLE COSTS IN \$K

R & D ESTIMATE	1000.00
INVESTMENT-NON-RECURRING	2808.00
INVESTMENT-RECURRING	7020.00
TOTAL INVESTMENT	9828.00
OPERATIONS	85105.99
LOGISTICS SUPPORT	8890.44
OPERATIONS & LOGISTICS SUPP. TOTAL	93996.42
TOTAL LIFE CYCLE COST	104824.42

BEST AVAILABLE COPY

ANNUAL O & S COSTS IN \$

ELECTRIC POWER	21024.00
SPECIAL MATERIAL	90000.00
OPERATOR PERSONNEL	10800000.00
ORG. MAINT. PERSONNEL	0.00
INT. MAINT. PERSONNEL	87512.40
DEPOT MAINT. PERSONNEL	392942.47
TOTAL MAINT. PERSONNEL	480454.87
SUPPORT EQUIP. MAINT.	500.00
SOFTWARE MAINTENANCE	0.00
SUPPLY PERSONNEL	2625.37
SPARE PARTS & REPAIR MATERIAL	132144.60
INVENTORY MANAGEMENT	38250.00
INVENTORY HOLDING	265867.56
INVENTORY ADMINISTRATION	304117.56
TRANSPORTATION	6607.23

LIFE CYCLE COSTS IN \$K

R & D ESTIMATE	1000.00
INVESTMENT-NON-RECURRING	3258.72
INVESTMENT-RECURRING	8146.79
TOTAL INVESTMENT	11405.50
OPERATIONS	109110.24
LOGISTICS SUPPORT	9264.50
OPERATIONS & LOGISTICS SUPP. TOTAL	118374.74
TOTAL LIFE CYCLE COST	130780.24

Example 2

BEST AVAILABLE COPY

INPUT COST ELEMENT VALUES

R1	=	2920.00	R21	=	.05	R65	=	.0500
R2	=	20.00	R22	=	0.00	R66	=	.0100
R3	=	.50	R24	=	100.00	R67	=	40.00
R4	=	10.00	R25	=	3.00	R71	=	0.00
R5	=	15.00	R26	=	0.00	R72	=	5.00
R6	=	10.00	R27	=	.10	R75	=	0.00
R7	=	23.00	R28	=	5.00	R76	=	0.00
R8	=	.48	R29	=	5000.00	TF	=	B
R9	=	100.00	R50	=	0.00	R78	=	0.00
R10	=	3.00	R51	=	0.00	R79	=	0.00
R11	=	12000.00	R52	=	0.00	X(1)	=	1070.00
R12(1)	=	25.00	R53	=	0.00	X(2)	=	770.00
R12(2)	=	25.00	R54	=	0.00	X(3)	=	580.00
R12(3)	=	25.00	R55	=	0.00	X(4)	=	460.00
R12(4)	=	25.00	R56	=	0.00	Y(1)	=	720.00
R13	=	30.00	R57	=	1000000.00	Y(2)	=	420.00
R14	=	1000.00	R58	=	0.00	Y(3)	=	130.00
R15	=	3.123	R59	=	0.00	Y(4)	=	110.00
R16	=	1000.00	R60	=	5.00	UP	=	50000.00
R17	=	20.00	R61	=	5.00	R80	=	0.00
R18	=	30000.00	R62	=	30.00	R81	=	0.00
R19	=	234.00	R63	=	25.00	INVR	=	T
R20	=	0.00	R64	=	3000.00	INVN	=	1000000.00

ANNUAL O & S COSTS IN \$

ELECTRIC POWER	7008.00
SPECIAL MATERIAL	30000.00
OPERATOR PERSONNEL	3600000.00
ORG. MAINT. PERSONNEL	0.00
INT. MAINT. PERSONNEL	29170.80
IEPOT MAINT. PERSONNEL	223839.53
TOTAL MAINT. PERSONNEL	253010.33
SUPPORT EQUIP. MAINT.	500.00
SOFTWARE MAINTENANCE	0.00
SUPPLY PERSONNEL	875.12
SPARE PARTS & REPAIR MATERIAL	44048.80
INVENTORY MANAGEMENT	38250.00
INVENTORY HOLDING	167434.46
INVENTORY ADMINISTRATION	205684.46
TRANSPORTATION	47906.25

LIFE CYCLE COSTS IN \$K

R & D ESTIMATE	1000.00
INVESTMENT-NON-RECURRING	1000.00
INVESTMENT-RECURRING	5000.00
TOTAL INVESTMENT	6000.00
OPERATIONS	36370.08
LOGISTICS SUPPORT	5520.24
OPERATIONS & LOGISTICS SUPP. TOTAL	41890.32
TOTAL LIFE CYCLE COST	48890.32

Section 6. SPECIAL CONDITIONS

6.1 Arithmetic Execution Errors

An execution error will occur if, due to the presence of an unorthodox combination of input costs elements, a division by zero is attempted or the operand of an arithmetic operation becomes excessively large, causing an overflow condition.

Additionally, a non-positive value for the cost element R20 (learning curve slope(%)) causes the model to attempt to calculate a non-existent logarithm. An excessively large value for R20 (on the order of 10^9) also results in a non-determinable situation.

The presence of any of the above conditions (and possibly others) causes program execution to be terminated prematurely (cond (b), section 3.3). Consequently, the file of back-up data is catalogued in the system. The two-line message:

```
CT ID = LAI PFN=BACKUP
CT CY = 001 00000192 WORDS.:
```

informs the program operator that this file has been saved.

If the multiple family option has been selected and processing has advanced beyond the first member of the first family, a permanent file consisting of the multiple family data will be catalogued also, as indicated by the display:

```
CT ID: LAI PFN=MULTFAM
CT CY: 001 00001024 WORDS.:
```

The terminal will next display "COMMAND-" and the program may be restarted by responding with 'MCDEC.', followed by a carriage return. The course of action which the program subsequently takes is described in section 3.3.

6.2 Phone Line Loss

When the communication link between the terminal and the computer is lost during execution, the program continues running until another response from the operator is required or the end is reached. Therefore, in the event of a phone line loss, the operator should immediately attempt a reconnection by re-dialing the computer, then submitting the LOGIN command (section 3.1). What happens next depends on how much time has elapsed between disconnection and reconnection, as discussed in sections 6.2.1 and 6.2.2.

6.2.1 Fewer Than Five Minutes

If the operator has reestablished communication with the computer in fewer than five minutes and the end of the program has not been reached, the message "WAITING FOR INPUT" is displayed after the login. The operator must submit a response to the request which would have been issued next by the program had the disconnect not occurred, in order for normal execution to resume. This may require knowledge of the program's operating sequence (section 3) and could involve guesswork. If the results of a run were being printed when the interruption occurred, the undisplayed portion cannot be recovered.

6.2.2 More Than Five Minutes

When five minutes have passed since the disconnect, the system aborts the program if communication has not been reestablished, thereby causing the backup data (and possibly multiple family data) to be saved permanently on a file. The display of "COMMAND-" after the login indicates that this is the case. The program must be restarted by re-typing instructions #1 - #4 as given in section 3.1. The back up file(s) are then accessed, and the current run may be resumed at the point of interruption.

6.3 Permanent File Errors

Since permanent file functions (e.g. access, catalog, purge, etc.) are done from within the program, a set of messages are included which identify the various error conditions which can result from these attempts. The general form of the error messages issued is as follows:

PERMANENT FILE ERROR NUMBER **xxx** ON **yyy** ATTEMPT.
DATA FILE FOR **zzz**

where **xxx** - a 3-digit (octal) error code (see Table V)
 yyy - indicates which function (access, purge, return, extend, request or catalog) was unsuccessful
 zzz - indicates which data file (back up, algorithm, multiple family, save, or retrieve) was involved in the error

Saving of the back up data if an abort occurs will not take place subsequent to an error condition pertaining to this file. Likewise, an error concerning the multiple family file renders its recovery impossible following (further) abort conditions. Thus, multiple family sensitivity analysis cannot be continued when an error of this type occurs.

Two functional types (access and return) are applicable to the algorithm data files (section 4.9.3). If the file is inaccessible, the algorithm cannot be used and control is returned to the current input sequence. The cost element(s) which this algorithm defines is set to zero.

If the error occurs when the file was being returned, (presumably the algorithm is finished with the file for the time being), the cost element(s) which is involved remains unaffected. However, subsequent invocation of the same algorithm could result in an access error.

Not all of the error codes (Table V) are applicable to every functional error and every data file type. Additional information on the permanent file functions can be found in reference 3, section 5 (SCOPE ref. manual).

Table V
PERMANENT FILE ERROR CODES

<u>Code</u>	<u>Description</u>
004	No room for extra cycle
005	Permanent file catalog full
012	File not in system
013	Archive retrieval aborted
015	Cycle number limit reached
016	Permanent file directory full
023	Cycle incomplete
024	PF already attached
025	File unavailable
030	File dumped
035	File already in system

6.4 Interrupting the Terminal

6.4.1 Deliberate Abort

The user may abort the program while it is executing by depressing the CTRL and Z keys simultaneously, then immediately typing %A. This may only be done when the terminal is printing, however. Aborting the program in this manner causes the back up file(s) of data to be written (section 3.3). The user will be notified of the successful catalog function via the message described in section 6.1, and may then restart the program as detailed in section 6.1.

6.4.2 Discard Current Output

The output currently being transmitted to the terminal may be discarded by the user by simultaneously striking CTRL and Z followed immediately by %S. This may need to be done repeatedly when a large quantity of output is being printed. The operator must be aware of the fact that the next message displayed by the program, asking for a response, may be discarded also. The program requires a legal response to this un-

displayed message as the next action taken. Consequently, the operator must know what it would have been in order to respond properly.

6.5 Data Save/Retrieval Permanent File Maintenance

Indiscreet use of the program's data saving capability could result in unnecessary (and costly) cluttering of NSWC/DL's CDC 6700 permanent file system. Brief descriptions of system utilities which can help monitor and control permanent file usage are given in sections 6.5.1 - 6.5.2. A more detailed explanation can be found in reference 3.

6.5.1 Audit

An up-to-date list of the user's permanent files may be obtained by typing 'AUDIT' in response to the terminal's display of 'COMMAND-'. Each list entry includes the permanent file's name, cycle number, accounting code, expiration date, and size. An additional column headed by "ERRORS" is also furnished. An entry of 'X' in this column indicates that the file's expiration date has passed and that the file will soon be purged by a system permanent file clean up routine. A 'Z' in this column indicates that the accounting code for the file is out of money. Purging of the file by the system is again imminent.

Although the retention period for a saved data file is specified as one year at creation time, failure to access this file during any 30 day span will result in the file being transferred to a special archive tape by the system. An entry of 'ARC' in the expiration date column of the audit indicates that the file has been archived. The file must then be reinstated in the system before it can be used again. This can be done automatically by the system at retrieval time. However, the user should be aware that a long wait at the keyboard will probably be necessary if file recovery is performed in this manner. Archived files which are not restored within six months of the archive date are permanently purged from the system.

6.5.2 Purge

Permanent files which are no longer wanted can be removed completely from the system by entering (after the display of 'COMMAND-'):

PURGE, pfn, ID=XXX.

where

pfn - permanent file name

XXX - the user ID associated with the file, e.g.

'LAI' for MCDEC and 'LUV' for NAVELEX.

The terminal will display

PR ID = XXX PFN=perm. file name
PR CY = nnn 00000128 WORDS:

to indicate that the file has been successfully removed. Following each purge, the command

RETURN,AAAAAAA.

should be entered, where 'AAAAAAA' is replaced by the permanent file name of the file just purged or the first seven characters of the name, whichever is shorter.

The procedure given above may also be utilized when an unwanted back up file (sections 3.3, 6.1) is present. This file can be removed by entering the following:

PURGE, BACKUP, ID=XXX.
RETURN, BACKUP.

The multiple family back up permanent file may be similarly removed by entering:

PURGE, MULTFAM, ID=XXX.
RETURN, MULTFAM.

6.6 Miscellaneous Topics

6.6.1 Program File Already in Use

The permanent file on which the program resides may not be available at the time the access attempt is made (instruction #1, section 3.1). This is indicated by a message of the form:

WAITING FOR ACCESS TO FILE WAITING FOR PF UTILITY

When this happens the user must either wait until the file is available or hang up the phone and try again later.

6.6.2 Output of Oversize Values

It is possible for cost element values to become too large for the field width allotted for their display. This may occur during input (when the current value of a cost element is displayed) or during the output of the results (either the input cost element values or the computed costs). In any event, a row of asterisks (*) will be printed in the field instead of the value.

6.6.3 Error Summary

Illegal characters submitted as input are not fatal to program execution. However, the system keeps a record of the errors which occur and their corresponding frequencies. When the program is terminated normally, a listing of the errors which occurred during the run are displayed in the form:

ERROR SUMMARY	
ERROR	TIMES
xxxx	yyyy

Where xxxx - an error code which identifies the error (reference 4 (FTN ref. manual))
 yyyy - indicates the frequency of the error

6.6.4 Correct Typing Errors

Typing errors may be corrected (prior to the carriage return) by holding the CTRL key down and depressing H as many times as desired, then making the correction. The intercom terminal is effectively backspaced one character for each 'H', although this will not be apparent on some models.

REFERENCES

1. Cost Effectiveness Program Plan for Joint Tactical Communications,
Volume III, Life Cycle Costing.
2. Marine Corps TRI TAC Cost Model (unpublished)
3. SCOPE Reference Manual, Control Data Corporation pub. no. 60307200.

NOT
Preceding Page BLANK - FILMED

DISTRIBUTION

Commander (2)
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703
ATTN: DRSEL-CP-CA (Mr. Trigg)

U.S. Army Signal School (1)
MSD Division
Fort Gordon, Georgia 30905
ATTN: Mr. Marion Macuch

AAI (2)
1501 Wilson Boulevard
Arlington, Virginia 22209
ATTN: Dr. Royal Randall

Naval Electronic Systems Command (27)
Washington, D.C. 20360
ATTN: Ron Ashby
Code: 5041

Joint Tactical Communications Office (2)
Fort Monmouth, New Jersey 07703
ATTN: Capt. David J. Boyd
Operations Research Division

Marine Corps Development and Education Center (2)
C3 Division
Quantico, Virginia 22314

Defense Documentation Center (12)
Cameron Station
Alexandria, Virginia 22314

Defense Printing Service
Washington Navy Yard
Washington, D.C. 20374

Library of Congress (4)
Washington, D.C. 20540
ATTN: Gift and Exchange Division

Headquarters Marine Corps
Washington, D.C. 20380
ATTN: Major A. H. Mossler
Code: CCTS-614

DISTRIBUTION

DX-21	
DX-222	(2)
DX-40	(6)
DK-70	
DK-72	(2)
DF-10	(33)
DN-11	(Glazman)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR-3645	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER MODEL FOR LIFE CYCLE COSTING USER'S GUIDE	5. TYPE OF REPORT & PERIOD COVERED	
7. AUTHOR(s) Wayne McLaughlin	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Surface Weapons Center Dahlgren Laboratory Dahlgren, Virginia 22448	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE May 1977	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
16. DISTRIBUTION STATEMENT (of this Report)	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report provides instructions for operating the computer program which models life cycle costs for equipment acquisitions. The program, written in the Fortran IV Extended programming language, was implemented on the CDC 6700 computer under the SCOPE 3.4 operating system at the Naval Surface Weapons Center, Dahlgren, Virginia, in January, 1977.		
Designed for use from an intercom terminal in an interactive manner, the model has a wide variety of built in capabilities which can greatly facilitate		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CI

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

cont

→ the task of performing life cycle cost analysis. A detailed discussion concerning the use of each feature is presented. This is then expanded into an overall description of the program's operating sequence, to aid the potential user in ascertaining if the model can be used to meet his specific needs.

The remainder of the report is devoted to the actual operation of the program. Execution involves the sequential display of requests by the terminal for input, followed by responses typed by the operator. Thorough coverage is given to the choices available to the user in each situation, including examples of input and output.

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)